



TAMPEREEN  
AMMATTIKORKEAKOULU

# LASKUTUSJÄRJESTELMÄ ANGULARJS:LLÄ

Mikko Mustalahti

Opinnäytetyö  
Joulukuu 2015  
Tietotekniikka  
Ohjelmistotekniikka



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikka  
Ohjelmistotekniikka

MUSTALAHTI MIKKO:  
Laskutusjärjestelmä AngularJS:llä

Opinnäytetyö 23 sivua, joista liitteitä 0 sivua  
Joulukuu 2015

---

Tämä opinnäytetyö tehtiin yritykselle, joka tarvitsi uuden laskutusjärjestelmän. Työn tarkoituksena oli tehdä selaimessa toimiva laskutusjärjestelmä yritykselle sekä perehtyä AngularJS:n toimintaan ja laskutusjärjestelmään. Sovelluksen front-end tehtiin käyttämällä AngularJS:ää yhdessä HTML5:n kanssa ja back-end tehtiin PHP:lla. Sovellusta varten tehtiin myös tietokanta MySQL Workbench -ohjelmalla.

Ohjelmakoodin kirjoittaminen sujui välillä hitaasti, koska työn tekemisessä oli paljon uutta asiaa, mikä tarvitsi ensin sisäistää, jotta ohjelmakoodia pystyi kirjoittamaan. Sovelluksesta tuli responsiivinen ja helppokäyttöinen, mutta siitä olisi voinut tulla vieläkin helppokäyttöisempi. Sovellus toimi hyvin testaamisessa käytettävän koneen selaimella, mutta usealla laitteella toimimista ei valitettavasti vielä ehditty tässä työssä testaamaan.

Laskutusjärjestelmä saatiin toimivaksi, mutta ajatuksena on sitä vielä jatkokehittää ennen kuin se otetaan käyttöön. Lisäksi sovellukseen on tarkoitus tehdä vielä käyttöohjeet, vaikka sen käyttäminen on jo nyt helppoa.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Computer Science  
Software Engineering

**MUSTALAHTI MIKKO:**  
Invoicing System with AngularJS

Bachelor's thesis 23 pages, appendices 0 pages  
December 2015

---

This thesis was done for a company that needed a new invoicing system. The purpose of this thesis was to make a functional invoicing system that works in a browser for the company, as well as became familiar with AngularJS's operations and invoicing system. The application's front-end was made using AngularJS together with HTML5 and back-end was made with PHP. Database was also made for the application and it was made using MySQL Workbench.

Writing code was occasionally slow because a lot of new things needed first to be internalized, so that writing the code was made possible. The application became responsive and easy to use, but it could have become even easier to use. The application worked well in testing device's browser, but unfortunately there wasn't enough time to test it on several devices yet.

The invoicing system was made functional, but the idea is to develop it even further before it is taken into use. In addition, instruction for use are intended to be made for the application, even though the use is already easy.

---

Key words: AngularJS, invoicing system, browser

## SISÄLLYS

1	JOHDANTO.....	6
2	MYYNTELASKUTUSJÄRJESTELMÄ JA SEN VAATIMUKSET .....	7
	2.1 Nykyinen järjestelmä .....	7
	2.2 Uuden järjestelmän vaatimukset.....	7
3	KÄYTETYT OHJELMISTOT JA KOODIKIELET .....	8
	3.1 AngularJS ja sen ominaisuudet.....	8
	3.1.1 Yhden sivun sovellus .....	8
	3.1.2 MVC-arkkitehtuuri.....	9
	3.1.3 Tietojen sidonta (Data Binding).....	10
	3.2 MySQL Workbench -ohjelma ja sen rooli.....	11
	3.3 HTML5, PHP ja Bootstrap .....	12
4	UUDEN JÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS .....	13
	4.1 Suunnittelun periaatteet .....	13
	4.2 Toteutus .....	13
	4.3 Testit ja niiden tulokset.....	20
5	POHDINTA.....	21
	LÄHTEET.....	23

**ERITYISSANASTO**

AngularJS	JavaScript-ohjelmistokehys
back-end	Sovelluksen palvelinpuolen toteutus.
Bootstrap	Verkkosivujen tekemisessä käytettävä HTML, CSS ja JavaScript -kehys.
front-end	Sovelluksen käyttäjälle näkyvä asiakaspuolen toteutus.
HTML	Hypertext Markup Language, Web-sivujen tekemisessä käytettävä kieli.
Javascript	Dynaaminen ohjelmointikieli
jQuery	JavaScript-kirjasto
MVC	Model-view-controller (malli–näkö–käsittelijä). Suunnitelumalli käyttöliittymän toteuttamiseksi.
MySQL	Relaatiotietokantaohjelmisto
PHP	Palvelinpuolen ohjelmointikieli
SPA	Yhden sivun sovellus (single-page application)
SQL	Kyselykieli, jolla voidaan tehdä relaatiotietokantaan erilaisia hakuja ja muutoksia.

## 1 JOHDANTO

Pienyrityksellä oli tarve uusia laskutusjärjestelmä, koska vanha ei enää ollut riittävä vaan tarvittiin lisää toimintoja. Haluttiin myös, että järjestelmää voisi käyttää monella eri laitteella.

Työn tarkoituksena oli tehdä toimiva laskutusjärjestelmä, joka on responsiivinen, helpokäyttöinen ja selaimessa toimiva. Laskutusjärjestelmän piti pystyä tallentamaan ainakin yrityksen tuotteet/palvelut (ja niiden hinnat yms.), yritysrekisterin (eli yrityksen asiakkaiden tiedot) ja tehdyt laskut. Laskutusjärjestelmästä piti myös tehdä helposti laajennettavissa oleva, jotta tulevaisuudessa sitä voisi käyttää usea yritys.

Erilaisista ohjelmistovaihtoehdoista päädyttiin AngularJS:ään, koska sillä tehtyä koodia voi suoraan käyttää monella eri laitteella ja haluttiin oppia käyttämään sitä. Lisäksi päädyttiin käyttämään nykyisin yleisiä MySQL Workbench -ohjelmaa ja PHP-koodauskieltä, koska niitä osattiin käyttää jo kohtalaisesti ja haluttiin erityisesti syventyä AngularJS:n käyttöön ja toiminnallisuuteen. Tallennettavat tiedot pistetään tietokantaan, joka tehtiin mainitulla MySQL Workbench -ohjelmalla. Sovelluksen front-end tehtiin AngularJS:llä ja HTML5:llä ja back-end PHP:llä.

## 2 MYYNTILASKUTUSJÄRJESTELMÄ JA SEN VAATIMUKSET

### 2.1 Nykyinen järjestelmä

Nykyinen myyntilaskutusjärjestelmä hakee asiakastiedot taulukkolaskenta-tiedostosta. Muut tiedot kirjoitetaan suoraan laskupohjaan. Hinta määräytyy tapauskohtaisesti, koska myytävä palvelu hinnoitellaan aina erikseen. Laskuja liikkuu kuukaudessa noin 15 kappaletta.

### 2.2 Uuden järjestelmän vaatimukset

Laskutusjärjestelmän vaatimuksista keskusteltiin työnantajan kanssa ja päädyttiin seuraaviin vaatimuksiin:

- Sovelluksen on oltava responsiivinen ja helppokäyttöinen.
- Sovelluksen on toimittava Internet-selaimessa ja usealla laitteella.
- Tehdään tietokanta ja tallennetaan sinne tarvittavat tiedot kuten yrityksen tuotteet ja palvelut, asiakasrekisteri ja tehdyt laskut laskutietoineen.
- Sovelluksen on pystyttävä hakemaan, lisäämään, muokkaamaan ja poistamaan tietokannan tietoja.
- Sovelluksesta pitää pystyä tulostamaan tehty lasku.
- Sovelluksen käyttöön tarvitaan käyttäjätunnukset. Salasana salataan.

Järjestelmästä haluttiin sellainen, että sitä voisi käyttää tulevaisuudessa myös erilaiset muut yritykset. Tyypillinen käyttötilanne on uuden laskun tekeminen, jossa käyttäjä valitsee tietokannassa olevan asiakkaan, lisää laskuun tuotteita/palveluja ja tämän jälkeen tallentaa ja tulostaa laskun.

### 3 KÄYTETYT OHJELMISTOT JA KOODIKIELET

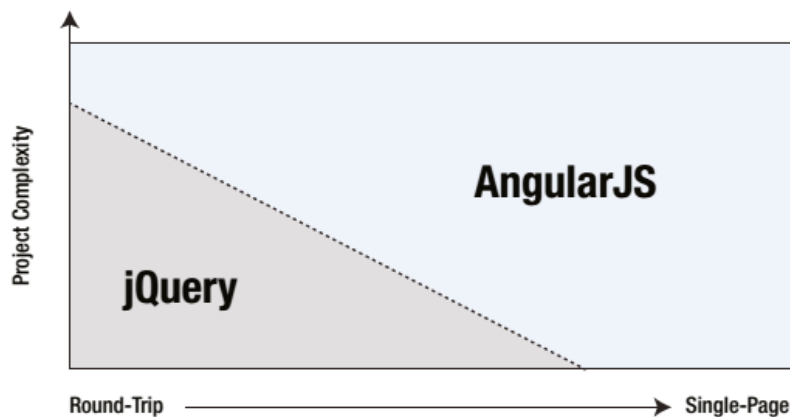
#### 3.1 AngularJS ja sen ominaisuudet

AngularJS on Javascript-ohjelmistokehys (framework). AngularJS on yhteensopiva sekä pöytäkoneiden että mobiililaitteiden selainten kanssa. (AngularJS 2015a.)

AngularJS on rakenteellinen kehys dynaamisille web-sovelluksille. Sen avulla voi käyttää HTML:ää mallin kielenä ja laajentaa HTML:n syntaksia sovelluksen komponenttien ilmaisemiseen selvästi ja ytimekkäästi. AngularJS:n tietojen sidonta (data binding) ja riippuvuusinjektio (dependency injection) vähentävät tarvittavan koodin kirjoittamisen määrää. Kaikki tämä toimii selaimessa, joka tekee sen ihanteelliseksi kumppaniksi minkä tahansa palvelinteknologian kanssa. (AngularJS 2015b.)

##### 3.1.1 Yhden sivun sovellus

Yhden sivun sovelluksissa alustava HTML-asiakirja lähetetään selaimelle ja sitä ei ikinä ladata uudestaan tai korvata. AngularJS antaa suurimman tuoton alkuperäisestä työtaakasta kun sovellus pääsee lähemmäs yhden sivun mallia. AngularJS:ää voi käyttää myös edestakaisissa sovelluksissa, mutta on muita teknologioita kuten jQuery, jotka sopivat paremmin niihin. Seuraavassa kuviossa nähdään millaisissa sovelluksissa AngularJS antaa hyötyä. (Freeman 2014, 46.)

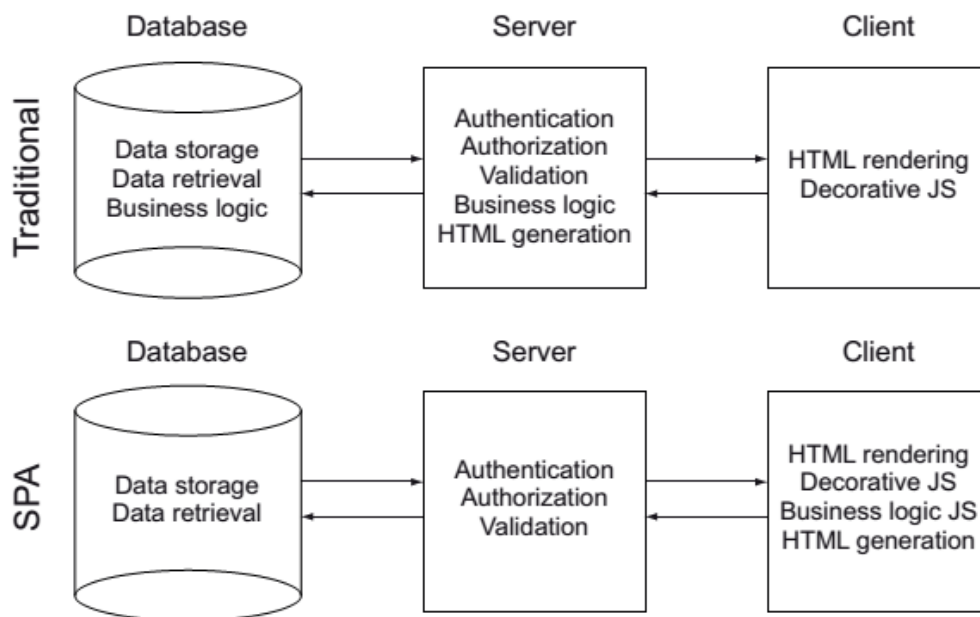


KUVIO 1. AngularJS sopii hyvin yhden sivun sovelluksiin (Freeman 2014, 46)



AngularJS loistaa yhden sivun sovelluksissa ja erityisesti monimutkaisissa monen html-sivun sovelluksissa. Yksinkertaisemmissa projekteissa jQuery tai vastaava vaihtoehto on yleensä parempi, mutta mikään ei estä käyttämästä AngularJS:ää kaikissa projekteissa. Nykyisillä web-sovelluksilla on vähittäinen taipumus siirtyä yhden sivun sovelluksen suuntaan ja se on AngularJS:n ihanne piste. (Freeman 2014, 46.)

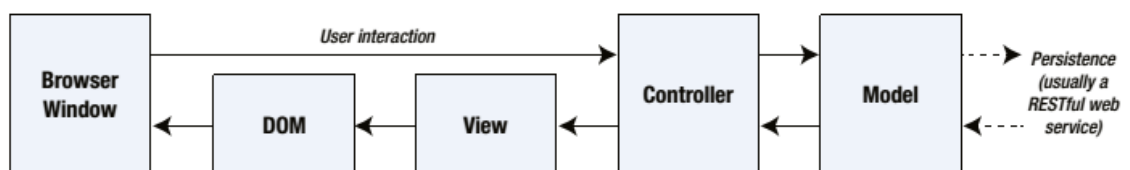
Kuviossa 2 kuvataan kuinka logiikka ja HTML:n luonti siirtyy palvelimelta asiakkaalle yhden sivun sovelluksissa.



KUVIO 2. Tietokannan (database), palvelimen (server) ja asiakkaan (client) velvollisuudet (Mikowski & Powell 2014, 8)

### 3.1.2 MVC-arkkitehtuuri

AngularJS toimii selaimessa, joka johtaa käänteeseen MVC-mallissa. Kuviossa 3 kuvataan, miten MVC-malli on toteutettu AngularJS:ssä. (Freeman 2014, 48.)



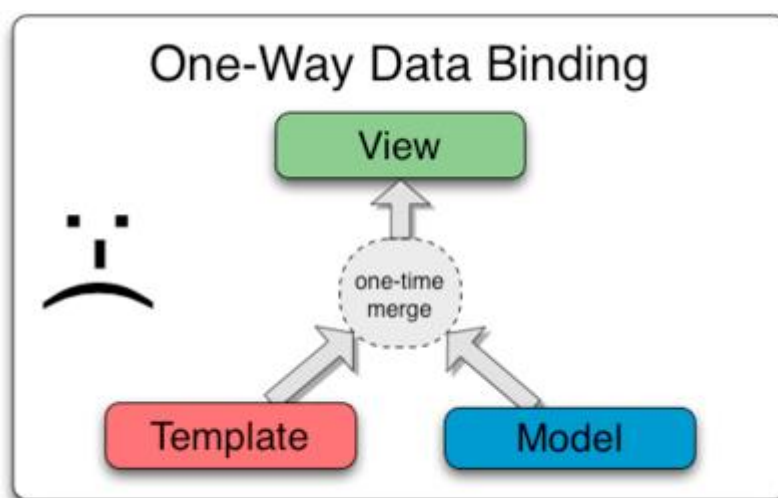
KUVIO 3. MVC-mallin toteutus AngularJS:ssä (Freeman 2014, 48)

MVC-mallin asiakaspuolen toteutus saa tietonsa palvelinpuolen komponenteilta. Käsitteilyjän (controller) ja näkymän (view) tavoitteena on käyttää mallin (model) tietoja suorittaakseen DOM-käsittelyä kuten niiden HTML-elementtien luonti ja hallitseminen, joiden kanssa käyttäjä voi vuorovaikuttaa. Nämä vuorovaikutukset syötetään takaisin ohjaimelle sulkien silmukan ja muodostaen vuorovaikutteisen ohjelman. (Freeman 2014, 48.)

### 3.1.3 Tietojen sidonta (Data Binding)

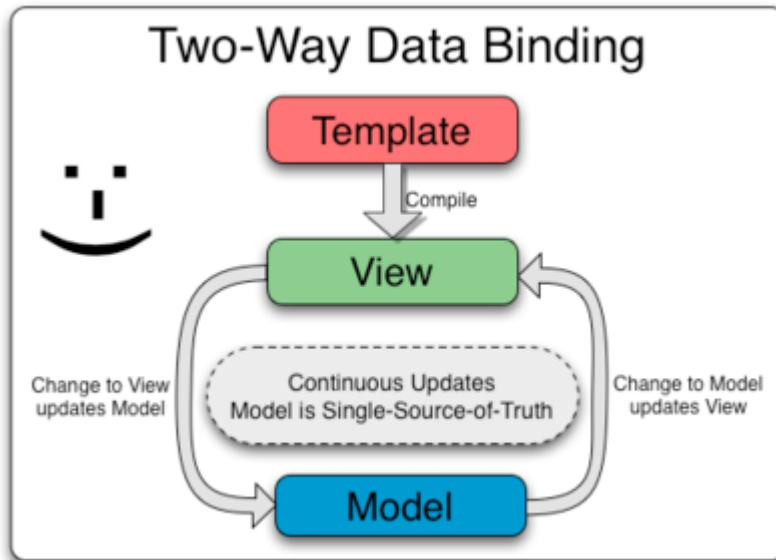
Tietojen sidonta Angular-sovelluksessa on tietojen automaattinen synkronointi mallin ja näkymän komponenttien välillä. Näkymä on projektio mallista koko ajan. Kun malli muuttuu, näkymä kuvastaa muutoksen ja toisin päin. (AngularJS 2015c.)

Useimmat mallijärjestelmät (templating systems) sitovat tiedot vain yhteen suuntaan (kuvio 4). Ne yhdistävät mallijärjestelmän ja mallin komponentit yhteen näkymään. Yhdistämisen jälkeen muutoksia malliin tai siihen liittyviin näkymän osiin ei automaattisesti näytetä näkymässä. Mitkään käyttäjän tekemät muutokset näkymään eivät myöskään automaattisesti näy mallissa. Tämä tarkoittaa, että kehittäjän täytyy kirjoittaa koodi, joka jatkuvasti synkronoi näkymän mallin kanssa ja mallin näkymän kanssa. (AngularJS 2015c.)



KUVIO 4. Tietojen sidonta klassisessa mallijärjestelmässä (AngularJS 2015c)

Angular-mallit (template) toimivat eri tavalla (kuvio 5). Ensin malli (joka on kääntämätön HTML sekä muut merkkaukset ja direktiivit) käännetään selaimessa. Kääntäminen tuottaa live-näkymän. Mitkä tahansa muutokset näkymään näytetään heti mallissa ja mitkä tahansa muutokset malliin välitetään näkymään. Näkymää voidaan ajatella välittömänä mallin projektiona. (AngularJS 2015c.)



KUVIO 5. Tietojen sidonta Angular:ssa (AngularJS 2015c)

Koska näkymä on vain projektio mallista, käsittelijä on täysin erillään näkymästä ja tietämätön siitä. Tämän takia testaus on helppo tehdä, koska on helppoa testata käsittelijä eristettynä ilman näkymää ja liittyvä DOM/selain itsenäisesti. (AngularJS 2015c.)

### 3.2 MySQL Workbench -ohjelma ja sen rooli

MySQL Workbench on visuaalinen tietokannan suunnittelutyöväline, joka yhdistää SQL:n kehityksen, hallinnan, tietokannan suunnittelun, luomisen ja ylläpidon yhteen integroituun kehitysympäristöön MySQL-tietokantajärjestelmälle (Wikipedia 2015). Tällä ohjelmalla luotiin tietokanta laskutusjärjestelmää varten sekä käytettiin tietokannassa olevien tietojen katsomiseen ja muokkaamiseen tarvittaessa.

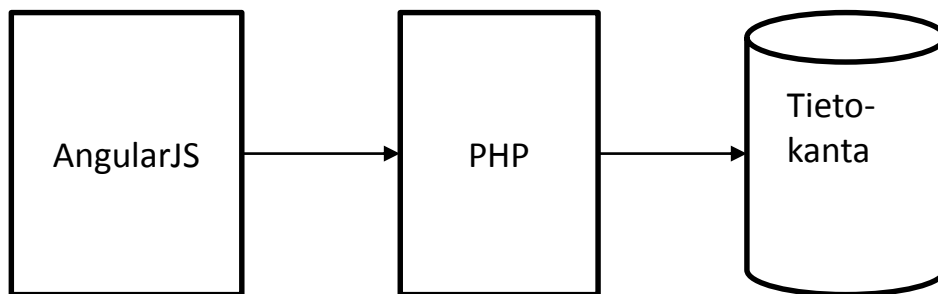
### 3.3 HTML5, PHP ja Bootstrap

Laskutusjärjestelmän tekemisessä käytettiin asiakaspuolella AngularJS:n kanssa HTML5:tä näytölle tulevien lomakkeiden ja muiden tietojen näyttämiseksi, koska ne sopivat hyvin yhteen. PHP:tä taas käytettiin SQL-käskyjen lähettämiseen ja muuhun palvelinpuolen koodiin. Bootstrap:ia käytettiin näkymän ulkoasun muokkaamisessa.

## 4 UUDEN JÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

### 4.1 Suunnittelun periaatteet

Sovellus tehtiin AngularJS:n toimintaperusteiden pohjalta yksisivuisena sovelluksena (SPA). AngularJS hoitaa sovelluksen näkymien näyttämisen yhdessä HTML5:n kanssa sekä toimintalogiikan. AngularJS lähettää http-pyyntöjä palvelimen PHP-tiedostoihin, jotka sisältävät SQL-käskyt, joilla tietokannan tietoihin pääsee käsiksi. Tätä kuvataan kuviossa 6.

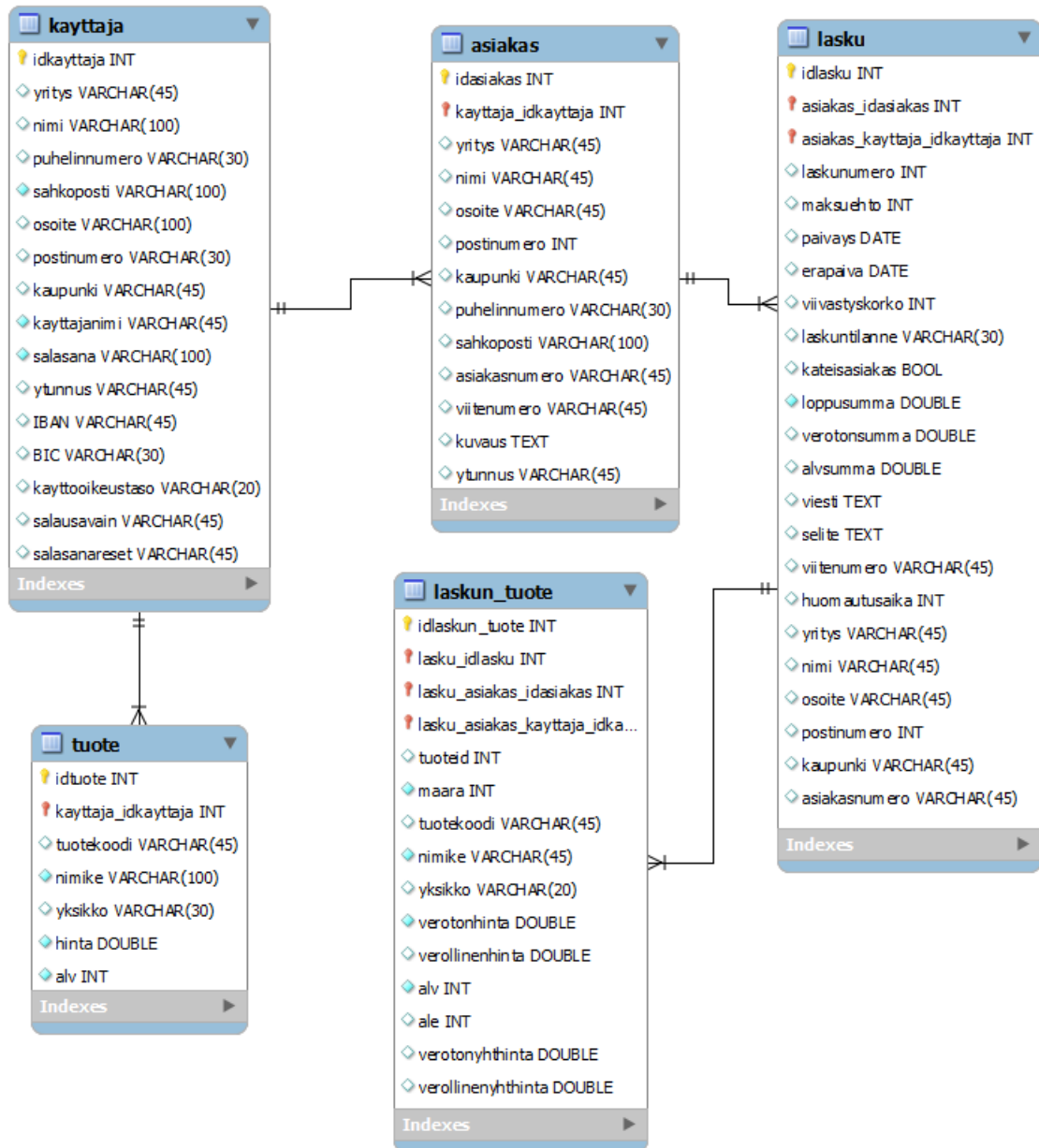


KUVIO 6. Toimintaympäristön rakenne

Sovelluksen käyttäjä tarvitsee käyttäjätunnukset, jotka tallennetaan tietokantaan salasana salattuna, jottei kukaan muu pääse kirjautumaan, vaikka tietokannassa olevat tiedot saataisiin jotenkin selville.

### 4.2 Toteutus

Ensimmäisenä tehtiin MySQL Workbench -ohjelmalla tietokanta sivuston käyttöä varten. Tietokantaan lisättiin tulevaisuutta varten myös sellaisia kenttiä, joihin ei tällä hetkellä tallenneta tietoja. Tietokannan rakenne näkyy kuvassa 1.



KUVA 1. Tämänhetkisen tietokannan ER-kaavio

Kun tietokanta saatiin valmiiksi, alettiin tekemään varsinaista sivustoa. Ensiksi tehtiin uuden asiakkaan ja uuden tuotteen lisäämistä varten lomakkeet käyttäen AngularJS:ää yhdessä HTML:n kanssa, jonka jälkeen alettiin koodaamaan tietojen tietokantaan lisäämisen mahdollistavaa koodia, jossa käytettiin myös PHP:tä. Samalla tehtiin myös taulukot asiakasrekisterin ja tuotelistan näyttämiseen. Kun tietojen lisääminen saatiin toimimaan, alettiin koodaamaan myös tietojen lukemista tietokannasta sekä tietojen muokkausta ja poistoa.

Seuraavaksi tehtiin lomake uuden laskun tekemiseen ja koodi näidenkin tietojen tallentamiseen, lukemiseen, muokkaamiseen ja poistamiseen. Vasta tämän jälkeen tehtiin oma

sivu, jolla näytetään itse lasku ja josta saadaan tulostettua näkymä ilman siinä näkyviä painikkeita. Sitten tehtiin vielä omat sivut laskun tietojen ja omien tietojen muokkaamista varten. Viimeiseksi tehtiin vielä yksinkertainen kirjautumissivu, jossa kysytään käyttäjätunnusta ja salasanaa. Tämän jälkeen tehtiin vielä koodiin korjauksia ja muokkaamista paremmaksi.

Uuden asiakkaan lisäämiseen ei vaadita muuta kuin asiakkaan tai yrityksen nimi, muut lomakkeen kentät ovat vapaaehtoisia. Vaadittavien kenttien reunat ovat punaiset ja asiakkaan lisäämisen tekevä painike ei ole toiminnassa ennen kuin yritys- tai nimi-kentässä on merkkejä (kuva 2). Samalla sivulla näytetään asiakasrekisteri, jossa lisätty asiakas näkyy heti lisäämisen jälkeen. Asiakkaan osoitetietoja ei kuitenkaan näy tässä listassa, koska haluttiin näkyviin vain tärkeimmät tiedot. Asiakasrekisteristä pystyy etsimään hakusalla ja asiakkaat pystytään lajittelemaan eri kenttien mukaan. Lisäksi sivulla on myös painikkeet, joista pääsee muokkaamaan asiakkaan tietoja ja poistamaan kyseisen asiakkaan. Asiakkaan poistamisessa on ponnahdusikkunalla tehty varmistus.

Etusivu
Luo uusi lasku
Luodut laskut
Tuotteet
Asiakasrekisteri
Omat tiedot
Kirjaudu ulos

### Uusi asiakas

asiakasnumero	yritys	nimi	osoite	postinumero	kaupunki
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
puhelinnumero	sähköposti	viitenumero	y-tunnus	kuvaus	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Lisää asiakas

### Asiakasrekisteri

Etsi:  Lajittele: Yrityksen mukaan ▾

asiakasnumero	yritys	nimi	puhelinnumero	sähköposti	viitenumero	y-tunnus	kuvaus		
		Pitkäyntysnimittä Oy	0501234567	pitkasähköpostiosoite@pitkaosoite.fi	234 32432 43243 23432	142548965-6		<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
34535	dhgsfdg	sdg						<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
765	fdg	gfdgdgsd					dfgdfskugf fdsgghdsjghf gdfghkfdshgdg sfgdhfkgdsd gkfsdhgdfsg dksgdfg	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
	gdfgs	sdgsdg						<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
234	gsdf	sdgsh sfg						<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
	gsfgfg	sgfsgdfgs						<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>
	harjoitus					9999977777		<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Muokkaa</span>	<span style="background-color: #0056b3; color: white; padding: 2px 5px; border-radius: 3px;">Poista</span>

KUVA 2. Tämänhetkinen asiakasrekisteri-sivu

Uuden tuotteen lisäämiseen vaaditaan vähintään tuotteen nimi, veroton hinta ja alv-prosentti (kuva 3). Kun hinta ja alv on kirjoitettu, ohjelma näyttää myös verollisen hinnan. Samalla sivulla on tuotelista, joka on tehty samaan tapaan kuin asiakasrekisterikin, eli

pystyy etsimään hakusanalla, lajittelemaan eri kenttien mukaan, sekä muokkaamaan ja poistamaan tuotteita.

[Etusivu](#)
[Luo uusi lasku](#)
[Luodut laskut](#)
[Tuotteet](#)
[Asiakasrekisteri](#)
[Omat tiedot](#)
[Kirjaudu ulos](#)

### Uusi tuote

tuotekoodi	nimike	ä-hinta veroton	ALV%	yksikkö	ä-hinta verollinen
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Lisää tuote](#)

### Tuotelista

Etsi:  Lajittelu: Nimen mukaan

tuotekoodi	nimike	ä-hinta veroton	ALV%	yksikkö	ä-hinta verollinen		
	asgjfdh sdgfgs dfgfdhg	128,00 €	24		158,72 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	atesti	234,00 €	23		287,82 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	balettitekko	150,00 €	24	kpl	186,00 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
alv10	hintatesti	100,00 €	10		110,00 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
alv13	hintatesti2	45,67 €	13		51,61 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	kaappi	35,00 €	24		43,40 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	marimekko paituli	25,00 €	24		31,00 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	porsaanfile	18,00 €	14	kg	20,52 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	sdfs	51,00 €	24		63,24 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>
	sfsdf	43,00 €	24		53,32 €	<a href="#">Muokkaa</a>	<a href="#">Poista</a>

KUVA 3. Tämänhetkinen tuotteet-sivu

Uuden laskun tekemiseen tarvitaan asiakasrekisterissä oleva asiakas. Pudotusvalikosta valitaan asiakas joko yrityksen tai nimen perusteella ja suodatinta käyttämällä pudotusvalikossa näytetään vain hakusanan sisältävät asiakkaat. Laskun numero on automaattisesti aina yhtä suurempi kuin aikaisempien laskujen suurin laskunumero, mutta sen voi itse halutessaan vaihtaa muuksi. Laskun päivämääräksi tulee automaattisesti nykyinen päivämäärä, mutta sen voi myös itse vaihtaa. Viivästyskorko on aina aluksi 8 % ja maksuehto 14 päivää, mutta myös nämä arvot voidaan itse vaihtaa. Laskun eräpäivä lasketaan päivämäärän ja maksuehdon perusteella. Selite-kenttä on täysin vapaaehtoinen.

Laskun tekemiseen tarvitaan lisäksi vähintään yksi tuoterivi. Laskulle voi lisätä tuoterekisterissä olevia tuotteita pudotusvalikosta ja suodattaa siinä näkyvät tuotteet. Lisäksi laskulle voi lisätä tyhjän rivin ja kirjoittaa itse tuotteen tiedot. Jos tuotteen lisää tuoterekisteristä, niin laskulle tulevia tuotteen tietoja voi silloinkin muokata tässä näkymässä, mutta muutokset eivät tallennu tuoterekisteriin. Tuotteiden määrä pitää muistaa pistää aina oikeaksi, sekä mahdollinen alennusprosentti. Laskulle tulevia tuoterivejä voi tietenkin



myös poistaa, jos esimerkiksi valitaan vahingossa väärä tuote. Sivulla näytetään myös laskun veroton ja verollinen yhteissumma sekä arvonlisäveron yhteissumma. Kun tarvittavat tiedot ja tuotteet on lisätty, lasku luodaan Luo lasku-painikkeella. Uuden laskun tekemistä havainnollistetaan kuvassa 4.

[Etusivu](#) [Luo uusi lasku](#) [Luodut laskut](#) [Tuotteet](#) [Asiakasrekisteri](#) [Omat tiedot](#) [Kirjaudu ulos](#)

### Uusi lasku

Valitse asiakasrekisterissä oleva yritys tai nimi. Jos asiakasta ei ole vielä asiakasrekisterissä, pääset lisäämään sen tästä linkistä.

Suodatin:

Yritys:  Nimi:  Tarvitaan yritys/nimi!

Laskun numero:

Selite:

Laskun päivämäärä:

Maksuehto:  päivää netto

Laskun eräpäivä:

Viivästyskorko:  %

Valitse alusvetovalikosta tuotelistassa oleva tuote/palvelu tai lisää laskuun uusi tuote/palvelu. Tällä sivulla tuotteisiin tehdyt muutokset eivät tallennu tuotelistaan. Tästä linkistä voit lisätä tuotteita sinne.

tuotekoodi	nimike	ä-hinta veroton	ä-hinta verollinen	ALV %	määrä	yksikko	ale %	yhteensä veroton	yhteensä verollinen	
	tossut	39	48,36	24	1		0	39	48,36	Poista
	kaappi	35	43,4	24	1		0	35	43,4	Poista

Suodatin  Lisää tuote  Luo lasku

Veroton yhteensä: 74,00 €  
 ALV: 17,76 €  
 Loppusumma: 91,76 €

KUVA 4. Tämänhetkinen sivu uuden laskun lisäämiseen

Kun lasku on luotu, näytetään kuvassa 5 oleva näkymä. Tulosta-painikkeesta saa näkyvässä olevan laskun tulostettua ilman näkyvässä olevia painikkeita. Jos laitteella on asennettuna ohjelma, jolla voidaan katsella pdf-tiedostoja, niin todennäköisesti asennettuna on myös ajurit, joiden avulla tästä voidaan tulostaa lasku myös pdf-tiedostona. Tämä tehdään valitsemalla tulostimeksi asennettuna oleva pdf-tulostin.

Koodi	Nimike	Määrä	Yks	Ä-hinta veroton	Ä-hinta verollinen	ALV %	Ale %	Yhteensä veroton	Yhteensä verollinen
123	asgsfd	1	a	123,40 €	153,02 €	24	0	123,40 €	153,02 €
<b>Yhteensä:</b>								<b>153,02 €</b>	

Veroton yhteensä <b>123,40 €</b>	ALV yhteensä <b>29,62 €</b>	Loppusumma <b>153,02 €</b>
Viitenumero <b>43</b>	IBAN <b>FI1234567</b>	BIC <b>NDEAFIHH</b>

Testiyritys Oy Testitie 5 3242 Testii	Y-tunnus: 1234567-8	puh. 040 1234567
---	---------------------	------------------

[Tulosta](#)   [Muokkaa](#)

KUVA 5. Tämänhetkinen tehdyn laskun näkymä

Jos laskussa huomataan virhe, niin laskun tietoja pääsee muokkaamaan Muokkaa-painikkeesta. Kun muutokset on tehty, niin Tallenna muutokset -painikkeesta tallennetaan muutokset tietokantaan. Tämän jälkeen voidaan siirtyä Näytä lasku -painikkeesta laskunäkymään ja tulostaa lasku. Laskun muokkaamissivun näkymä on kuvassa 6.

tuotekoodi	nimike	ä-hinta veroton	ä-hinta verollinen	ALV %	määrä	yksikko	ale %	yhteensä veroton	yhteensä verollinen
123	asgsfd	123,4	153,02	24	1	a	0	123,4	153,02

[Suodatin](#)   [Lisää tuote](#)   [Lisää tyhjä rivi](#)   [Tallenna muutokset](#)   [Näytä lasku](#)

Veroton yhteensä: 123,40 €  
 ALV: 29,62 €  
 Loppusumma: 153,02 €

KUVA 6. Tämänhetkinen laskun muokkaamissivu

Luodut laskut-sivulla (kuva 7) on listattuna tietokannassa olevat laskut, joista näytetään vain tärkeimmät tiedot. Näytä-painikkeesta pääsee laskunäkymään, josta oli esimerkki kuvassa 5.

[Etusivu](#)
[Luo uusi lasku](#)
[Luodut laskut](#)
[Tuotteet](#)
[Asiakasrekisteri](#)
[Omat tiedot](#)
[Kirjaudu ulos](#)

### Luodut laskut

Etsi:  Lajittele: Uusin päiväys ensin

laskun numero	yritys	nimi	päiväys	eräpäivä	loppusumma			
569	harjoitus		2015-10-20	2015-11-03	837,64 €	Näytä	Muokkaa	Poista
566	yritys1	nimi1	2015-10-08	2015-10-22	42,78 €	Näytä	Muokkaa	Poista
567	yritys1	nimi1	2015-10-08	2015-10-22	63,30 €	Näytä	Muokkaa	Poista
568	yritys1	nimi1	2015-10-08	2015-10-22	85,56 €	Näytä	Muokkaa	Poista
565	harjoitus		2015-10-07	2015-10-21	110,00 €	Näytä	Muokkaa	Poista
563	Ihana Puoti		2015-04-20	2015-05-04	321,16 €	Näytä	Muokkaa	Poista
43	fdg	gfdgdgsd	2015-03-30	2015-04-13	759,95 €	Näytä	Muokkaa	Poista
559	yritys1	nimi1	2015-03-30	2015-04-13	85,64 €	Näytä	Muokkaa	Poista
560	hemma		2015-03-30	2015-04-13	249,08 €	Näytä	Muokkaa	Poista
561	harjoitus		2015-03-30	2015-04-13	330,00 €	Näytä	Muokkaa	Poista
550	Hulabaloo	Niina	2015-03-27	2015-04-10	110,00 €	Näytä	Muokkaa	Poista
551	Hulabaloo	Niina	2015-03-27	2015-04-10	249,28 €	Näytä	Muokkaa	Poista
552	Hulabaloo	Niina	2015-03-27	2015-04-10	186,00 €	Näytä	Muokkaa	Poista

KUVA 7. Tämänhetkinen luotujen laskujen listaus-näkymä

Omat tiedot-sivulla voidaan tehdä muutoksia omiin tietoihin esim. osoitteen tai puhelinnumeron muuttuessa (kuva 8). Myös nämä tiedot tallentuvat tietokantaan.

[Etusivu](#)
[Luo uusi lasku](#)
[Luodut laskut](#)
[Tuotteet](#)
[Asiakasrekisteri](#)
[Omat tiedot](#)
[Kirjaudu ulos](#)

### Omat tiedot

Yritys:	Testiyritys Oy
Nimi:	Testinimi
Puhelinnumero:	040 1234567
Sähköposti:	sahkoposti@sposti.fi
Osoite:	Testitie 5
Postinumero:	3242
Kaupunki:	Testii
Y-tunnus:	1234567-8
IBAN:	FI1234567
BIC:	NDEAFIHH

[Tallenna muutokset](#)

KUVA 8. Tämänhetkinen omat tiedot-sivu

### 4.3 Testit ja niiden tulokset

Sovelluksen toimintaa testattiin paikallisesti käyttämällä XAMPP-ohjelmaa palvelimen ajamiseen. Sovelluksen toimintaa testattiin pääasiassa Firefox-selaimella, koska työnantaja käytti sitä selainta. Sovelluksen toimintaa testattiin aina sen jälkeen, kun tehtiin sovelluksen logiikkaan muutoksia. Toiminnan testaus suoritettiin kirjoittamalla kenttiin tietoja tai jättämällä tyhjäksi osa kentistä ja tutkimalla, tuliko tietokantaan oikeat tiedot yms. Toimintaa testattiin myös tulostamalla näytölle käytettyjen muuttujien sisältöä sekä seuraamalla, tekikö ohjelma mitä pitikin. Aina välillä tapahtui koodausvirheitä, mutta lopulta saatiin ohjelman toiminnot toimimaan halutulla tavalla.

## 5 POHDINTA

Ohjelmakoodin kirjoittaminen sujui välillä todella hitaasti, koska työn tekemisessä tuli monta sellaista asiaa vastaan minkä tekeminen piti itseopiskella ensin. Suuri osa ongelmista liittyi AngularJS:n toimintaan, mutta myös PHP-koodin kanssa tuli välillä sellaisia virheitä, joiden kanssa meni paljon aikaa.

Sovelluksesta tuli responsiivinen ja helppokäyttöinen, mutta siitä olisi voinut tulla vieläkin helppokäyttöisempi. Sovellukseen tarvitsee myös tehdä vielä käyttöohjeet, vaikka sen käyttäminen helppoa onkin. Sovellus toimi hyvin testaamisessa käytettävän koneen selaimella, mutta usealla laitteella toimimista ei valitettavasti vielä päästy testaamaan. Luultavasti siinä ei suurempia ongelmia tule vastaan. Sovelluksen käyttämään tietokantaan pystyy tallentamaan halutut tiedot sekä hakemaan, muokkaamaan ja poistamaan tietoja. Sovellusta jatkokehittäessä täytyy kuitenkin vielä lisätä laskuihin ainakin tieto siitä, onko kukin lasku maksettu vai ei. Tietokannassa on jo valmiina paikka tämän tiedon tallentamista varten. Sovelluksesta pystyy tulostamaan tehdyt laskut, mutta laskun ulkonäkö saattaa vielä muuttua. Muiden sivujen ulkonäköä ainakin tullaan vielä muuttamaan. Sovelluksen käyttöön tarvitaan käyttäjätunnukset ja salasana salataan, koska sovellusta tullaan käyttämään Internetin kautta. Salasanan salaus saatiin tehtyä helpommin kuin ensin luultiin, koska siihen löytyi valmis ja turvallinen toiminto.

Testattaessa sovellusta huomattiin, että joissakin alasetoalikoissa ei näy kaikki valintamahdollisuudet siirtämättä näyttöä alaspäin. Samoin huomattiin, että tarvitsee lisätä tietokannan lasku-tauluun kenttiä, joihin tallennetaan käyttäjän tietoja, että saadaan vanhat laskut näkymään sellaisina kuin ne laskujen kirjoitushetkellä ovat, vaikka käyttäjän tiedot muuttuisivatkin. Tällä hetkellä käyttäjän laskulla näkyvät tiedot nimittäin muuttuvat kaikkiin laskuihin, jos niitä muuttaa. Sovelluksesta jäi vielä pois mahdollisuus tehdä lasku sellaiselle asiakkaalle, joka ei ole vielä tallennettuna tietokantaan. Tämä mahdollisuus tullaan siihen varmasti lähitulevaisuudessa lisäämään.

Tulevaisuudessa voitaisiin myös tutkia, pystyisikö ja kannattaisiko pistää laskun tekeminen, muuttaminen ja tehdyn laskun näyttäminen samalle sivulle. Lisäksi voitaisiin tutkia, pystyisikö sovelluksesta lähettämään laskun suoraan pdf-tiedostona asiakkaan sähköpostiin ilman, että se ensin tallennetaan pdf-muodossa käyttäjän laitteelle ja sitten lisätään se

itse sähköpostin liitteeksi. Asiakasrekisteri-sivulle voisi lisätä mahdollisuuden saada halutut sarakkeet piiloon/näkyviin ja vaikka tuotelistaan myös. Lisäksi voitaisiin tehdä painikkeet, joista saisi uuden tuotteen ja asiakkaan lomakkeet piiloon/näkyviin. Uutta laskua tehdessä voisi olla hyvä pystyä samalla sivulla lisäämään asiakkaita ja tuotteita tietokantaan. Tulevaisuudessa voitaisiin myös antaa käyttäjälle mahdollisuus vaihtaa nykyiset vakioasetukset erilaisiksi kuten automaattisesti lajitteleminen jonkun muun sarakkeen mukaan ja viivästyskoron automaattisesti tuleva määrä.

## LÄHTEET

AngularJS. 2015a. AngularJS:n nettisivut (Miscellaneous/FAQ). Luettu 17.3.2015.  
<https://docs.angularjs.org/misc/faq>

AngularJS. 2015b. AngularJS:n nettisivut (Developer Guide/Introduction). Luettu 23.3.2015. <https://docs.angularjs.org/guide/introduction>

AngularJS. 2015c. AngularJS:n nettisivut (Developer Guide/Data Binding). Luettu 24.3.2015. <https://docs.angularjs.org/guide/databinding>

Freeman A. 2014. Pro AngularJS. New York: Apress.

Mikowski, M. & Powell, J. 2014. Single Page Web Applications. New York: Manning Publications.

Wikipedia. 2015. Wikipedian nettisivut (MySQL Workbench). Luettu 7.12.2015.  
[https://en.wikipedia.org/wiki/MySQL\\_Workbench](https://en.wikipedia.org/wiki/MySQL_Workbench)